Securing the U.S. Transportation Command

Jeffrey M. Diewald, Kajal T. Claypool, Jesslyn D. Alekseyev, George K. Baah, Uri Blumenthal, Alfred Cilcius, William L. Pughe, Joseph A. Cooley, Robert K. Cunningham, Jonathan R. Glennie, Edward F. Griffin, and Patrick J. Pawlak

The U.S. Transportation Command moves soldiers, equipment, and supplies around the world to support U.S. military and disaster relief operations. To help ensure that this critical supply chain is functioning efficiently, Lincoln Laboratory is working with the command to develop a software architecture that will provide the command with an enterprise network with ample computational power, strong cyber security, and resiliency to attacks and disruptions.

You will not find it difficult to prove that battles, campaigns, and even wars have been won or lost primarily because of logistics.

— General Dwight D. Eisenhower

History is rich with examples that showcase the power of military logistics and its influence in the outcome of wars. Hannibal crossing the Alps with foot soldiers, horsemen, and elephants to gain a string of victories in central Italy between 218 and 204 BCE relied on logistical planning, cutting supply lines for Roman forces and seizing Roman supply depots [1]. The six years of the Battle of the Atlantic in World War II were a struggle to get a million tons of imported material to Britain every week, fighting German efforts to sink as many of the cargo ships as possible [2]. As the battle raged in the Atlantic Ocean, Allied bombers were destroying German access to oil refineries and synthetic fuel factories. By 1944, the Germans did not have enough fuel for aircraft to protect the oil facilities that remained or for the fleet of submarines that had caused so much damage in the Atlantic [3]. These are just two examples that demonstrate the effectiveness of a military strategy to disable an enemy's supply lines.

The United States' extraordinary and unique ability to rapidly project national power and influence are a direct result of its transportation command—the United States Transportation Command (USTRANSCOM). Uninterrupted and efficient operation of the U.S.

transportation supply chain is critical for ensuring the nation's ability to deploy forces for military actions or humanitarian aid and disaster relief. USTRANSCOM, whose transportation systems have evolved out of many separate distribution systems and programs, is now looking to consolidate and refactor these disparate components and to secure its computing and storage infrastructure. Lincoln Laboratory has been enlisted to help architect this next-generation USTRANSCOM enterprise.

The Laboratory's efforts are addressing fundamental operational and cyber security issues to improve the overall USTRANSCOM defensive posture and cyber visibility across the command. The goal is to facilitate the development of an enterprise that is robust, secure, and resilient to disruptions, whether from cyber attacks, geopolitical turmoil, meteorological events, or natural disasters.

USTRANSCOM: Background and Enterprise Needs

USTRANSCOM is one of nine unified commands for the Department of Defense, providing air, land, and sea transportation in times of peace and war. USTRANSCOM, established in 1987, serves as the single manager of the U.S. global defense transportation system, supporting troop deployment and sustainment, air refueling, medical evacuations, presidential movements, as well as humanitarian aid and disaster relief missions. In a typical week, USTRANSCOM executes roughly 1900 air missions, 25 ship movements, and 10,000 ground shipments across 75% of the world's countries [4]. All USTRANSCOM missions are conducted worldwide and employ military and commercial transportation assets coordinated through the three USTRANSCOM Transportation Component Commands: the Air Force's Air Mobility Command, which provides aerial refueling capabilities and air transport for people and supplies; the Army's Surface Deployment and Distribution Command, which plans and executes surface deliveries of supplies and equipment; and the Navy's Military Sealift Command, which directs sea transportation [5]. Seventy percent of these movements are subcontracted to commercial partners, such as Maersk, United Parcel Service, and small local shipping companies [6].

As shown in Figure 1, a typical mission is initiated by a request, including a set of movement requirements, from a combatant command (COCOM). A

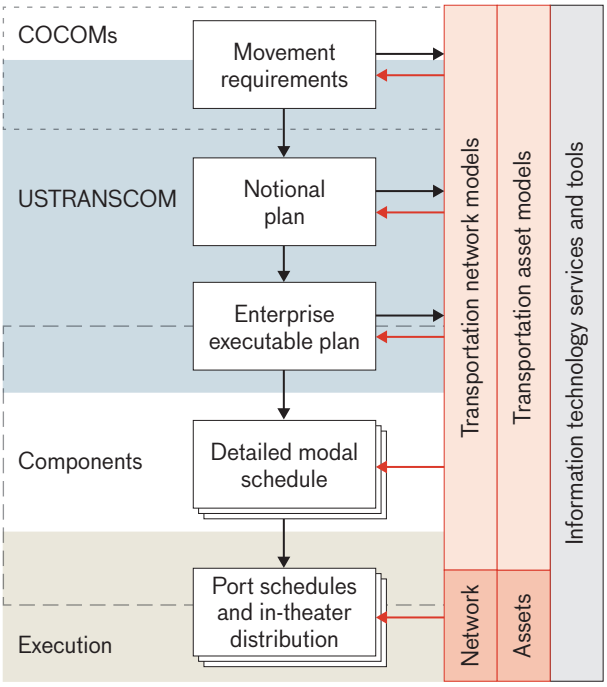


FIGURE 1. The USTRANSCOM operational flow begins with a request for a mission from a combatant command (COCOM) and progresses through the stages of planning and scheduling until the appropriate Transportation Component Command executes the mission, relying on the networks and assets available, including commercial carriers.

movement requirement specifies the type, quantity, source, destination, and timeframe for movements of goods and personnel. A notional plan is produced by USTRANSCOM Fusion Center personnel, in collaboration with the combatant and Transportation Component Commands [7, 8]. This plan is developed through an iterative process that evolves on the basis of the availability of the planes, ships, trains, and trucks needed to move goods and personnel from source to destination. The result of this process is an enterprise executable plan, which can be broken down into explicit instructions, i.e., a modal schedule, for each of the component commands.

This initial planning is sufficient to start the movements, but the world is an uncertain place, given the threat of cyber attack, the politics of military movements, the unpredictability of weather, the inevitability of mechanical failures, or the chaos created by a natural disaster. Thus, replanning is a fundamental process, perhaps the core process, in USTRANSCOM

movements. Ripples flow up and down the hierarchy in Figure 1; sometimes the higher-level planners can adjust their plans according to timely and accurate situational awareness, but often there is no time, and solutions must be found at the lower levels.

Because USTRANSCOM processes and distribution systems evolved separately and independently, coordinating and securing the operations of the various systems is a challenging task. Each command in the USTRANSCOM portion of the process has separate business processes and information systems, each with its own information representations, business rules, and constraints. Although these processes and systems are still functioning to accommodate USTRANSCOM's round-the-clock, all-year-long schedule of missions, they are inefficient and grow less secure as adversaries find and build new exploits to infiltrate computer networks. Differences in information systems necessitate individualized cyber security solutions to meet Department of Defense (DoD) security requirements and, therefore, increase the cost and effort needed to administer these systems. In addition, because the majority of its movements are executed by commercial vendors, USTRANSCOM is compelled to exchange information in schemas defined by those vendors and with information systems outside of the cyber security purview of the DoD. Many of these vendors are based outside the United States, run by foreign nationals who are operating their businesses across the open Internet.

Furthermore, USTRANSCOM's plans, as well as all the information underpinning those plans, are crucial to the United States; therefore, these plans are of great interest to U.S. adversaries. The most advanced of these adversaries are constantly probing for a foothold inside USTRANSCOM as part of their search for more permanent access to U.S. secrets. These adversaries also look to attack and compromise commercial vendors' systems as a pathway into the USTRANSCOM enterprise. The list of vendors targeted by cyber attackers includes cleared defense contractors that build and maintain applications used by USTRANSCOM. Adversaries hope that if they can compromise an application at a contractor's site, USTRANSCOM will not detect the exploitable capability inserted into a system and will then install it as part of a regular upgrade performed to synchronize USTRANSCOM systems with those of the contractor.

Any changes to USTRANSCOM systems, for modernization or enhanced cyber security, cannot delay ongoing missions. Nevertheless, USTRANSCOM recognizes that an incremental modernization and consolidation of their distributed architecture could bring significant improvements to the enterprise:

- Faster response to external events, improving the efficiency of operational plans and timelines
- More flexibility to overcome access challenges, such as bad weather, geopolitical uncertainties, and active anti-access/area denial efforts by adversaries
- Better throughput in wartime or crisis operations
- More efficient operations that would lower fuel expenses, contract costs, and maintenance costs for planes, ships, and other fleet vehicles
- Reduced data storage costs achieved by moving to the cloud or cloud-ready technologies
- Increased cyber security through a reduced, reproducible, consistent, and measurable cyber attack surface
- Improved cyber security for the software development supply chain

Several groups across two divisions at Lincoln Laboratory are collaborating to find architectural solutions that will allow USTRANSCOM to take advantage of these improvements. The Laboratory is providing these answers through prototypes, demonstrations, recommended technologies, and tactics, techniques, and procedures (TTP) that improve cyber security, all enabled by an architecture based on three key tenets:

- *Platform as a Service (PaaS)/Infrastructure as a Service (IaaS) lifecycle security*: Service lifecycle security focuses on defining system and software protections and visibility that should exist in the cloud to isolate malware and adversarial actions while maintaining resilient, visible operational systems. This effort has resulted in recommended technologies and TTPs that segment applications across the entire cloud-based software stack and that help counter the many threat vectors that exist in cloud computing. These recommendations include methods to keep data confidential, to guarantee data cannot be tampered with as they move from cloud to user, and to verify data only goes to authorized users. Lincoln Laboratory's approach has been to develop a high-assurance multitenant cloud environment that can be physically distributed across cloud infrastructures.

- *Data lifecycle security*: Data lifecycle security focuses on maintaining data protections and visibility while USTRANSCOM and third parties store, communicate, and manipulate enterprise data. This effort has led to the development and implementation of a strategy to maintain real-time visibility of the enterprise attack surface from the perspective of data as they flow across boundaries within and outside of USTRANSCOM. In particular, the Laboratory's strategy uses data provenance, which is a record of the history of the evolution of data in a computing system [9], for both understanding the critical enterprise data flows and protecting the data.
- *Authentication and authorization*: Each user and system is authenticated before being granted access to USTRANSCOM resources; this process is in keeping with the reference monitor idea first expressed by Anderson [10], and consistent with the National Institute of Standards and Technology's Identity and Access Management plan [11]. Whereas today USTRANSCOM uses a range of authentication and authorization approaches arising from the organic growth of the organization, Lincoln Laboratory is working to transition USTRANSCOM to an architecture that consistently and methodically provides protection. The Laboratory's approach focuses on the use of data provenance to automatically generate a consistent set of authentication and authorization security policies.

This article presents our development of the Lincoln Secure Environment (LSE), a private cloud hosted at Lincoln Laboratory and offering high-assurance PaaS and IaaS support for multiple tenants. We provide an overview of the threat model and security architecture of the LSE, which is part of the USTRANSCOM test range being implemented at Lincoln Laboratory and which serves as the development and demonstration environment for the Laboratory staff working on the USTRANSCOM project. The LSE also serves as a model for USTRANSCOM's software development environment.

The article also describes our work on Using Provenance To Expedite MAC Policies (UPTEMPO), a tool that uses collected data provenance for the generation of authentication and authorization security policies, and showcases UPTEMPO's use of data provenance to identify critical enterprise data flows and to generate mandatory access control (MAC) policies for improved access protection.

USTRANSCOM Next-Generation Architecture Overview

Figure 2 depicts the Laboratory's proposal for a next-generation USTRANSCOM architecture that provides a mature, secure information technology enterprise. The overall architecture has been developed with security-first principles; cyber security is integrated as a key driver of solutions within each layer.

This IaaS cloud is a high-assurance multitenant architecture that is designed to be vendor-agnostic and can be distributed across multiple physical infrastructures. The architecture consists of several layers, with clean, well-defined interfaces between them.

PLANNING AND ANALYSIS APPLICATIONS

At the very top layer are USTRANSCOM-specific applications, built with the business logic of the USTRANSCOM enterprise. These applications are the domain of planners, analysts, and cyber situational awareness at USTRANSCOM.[1]

SECURE NETCENTRIC ENTERPRISE BUS AND ITS SERVICES

This layer is a secure interoperable messaging system that provides a standard service-oriented architecture (SOA) that is federated across the USTRANSCOM enterprise. The message bus provides a simple, consistent interface to a wide range of small, common, sharable, and composable services that the applications can use to build their business logic. This layer of services also provides a federated and unified data-sharing environment for USTRANSCOM and its component commands. By breaking down the "stovepipes," i.e., the rigid implementation barriers that lock existing data in isolated databases, USTRANSCOM should be free to integrate existing data into new and useful combinations. The federated secure SOA provides the foundational global standard from which to support comprehensive interoperability to meet USTRANSCOM and its components' business needs.

This architectural approach has been successfully used at other government organizations. Lincoln

[1] Another large part of Lincoln Laboratory's USTRANSCOM project is experimenting with new planning algorithms to add robustness and resiliency for building plans. A third part of this USTRANSCOM project is developing new ways to capture and view cyber situational awareness information so as to detect adversaries earlier, minimizing their damage.

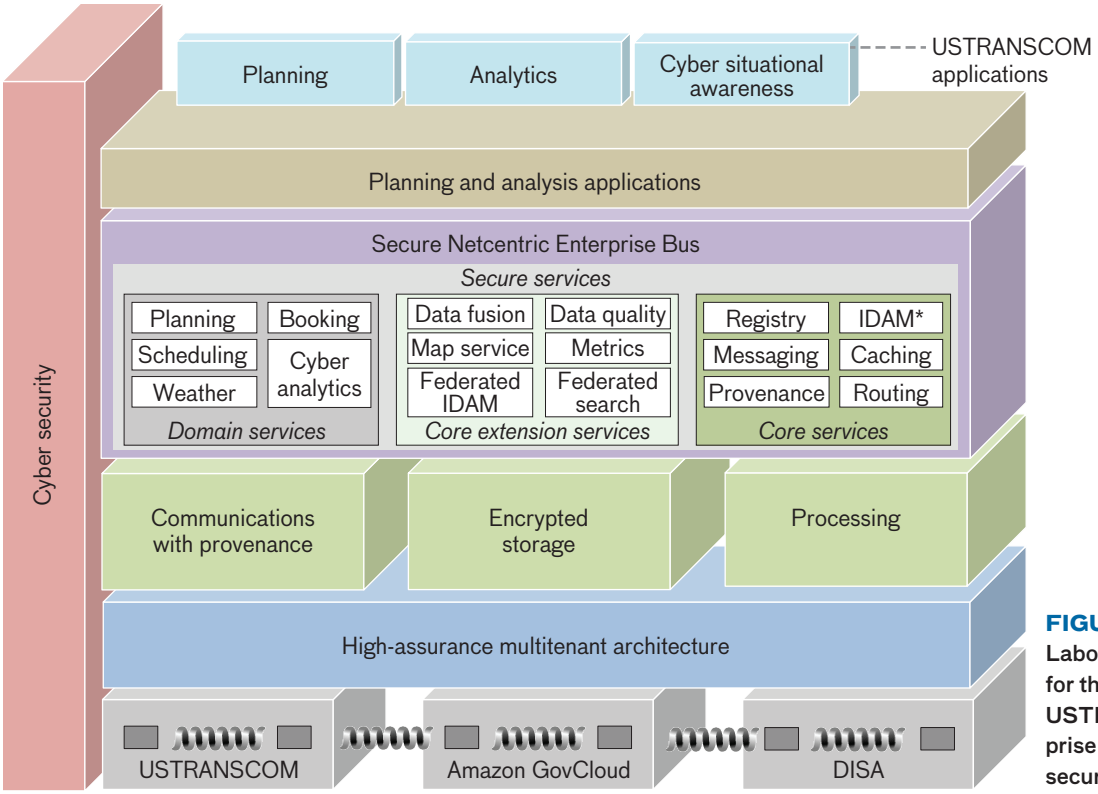


FIGURE 2. Lincoln Laboratory's proposal for the next-generation USTRANSCOM enterprise architecture builds security into every layer.

*IDAM–Identity and Access Management

Laboratory has used this design pattern at COCOMs for providing improved cyber situational awareness, at the Federal Aviation Administration to provide integrated weather information in support of human-in-the-loop decision support, and in several other Lincoln Laboratory projects [12–16].

FUNDAMENTAL SERVICES

All of the services above this layer share three common needs—communications, storage, and computational power—that are served by this layer. First, adding data provenance to the communication services will provide input for UPTEMPO (detailed in the section titled "Data Provenance in the Lincoln Secure Environment"), leading to stronger cyber security policies and a forensic trail for all communication paths. Second, ensuring that all data at rest are automatically encrypted adds another layer of protection against a determined adversary. Finally, harnessing the power of the cloud gives more processing power than before, allowing applications to experiment with new algorithms and techniques unavailable or impractical in conventional information technology environments.

HIGH-ASSURANCE MULTITENANT ARCHITECTURE

This layer provides a common interface to the underlying cloud infrastructures to allow for moving to a *hybrid cloud* model.[2] There are several cloud options currently available, each with a different set of security guarantees. While most of USTRANSCOM data are unclassified, some sensitive information is classified. Consequently, this classified information must stay within the confines of USTRANSCOM's private cloud. When the data are not sensitive, it should be possible to move that information into a more public cloud,[3] where USTRANSCOM can make use of the available economies of scale of commercial service providers.

[2] A hybrid cloud is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds) [17].

[3] The Amazon GovCloud is one example of a more public cloud that provides additional security assurances beyond those of the completely public Amazon Cloud offerings. GovCloud use is restricted to U.S. government projects.

A number of challenges must be addressed before this type of hybrid cloud infrastructure[4] can be become a reality. The article "Secure and Resilient Cloud Computing for the Department of Defense" in this journal highlights the current state of the art in this domain and the ongoing work at Lincoln Laboratory to address some of these challenges [18].

IAAS PHYSICAL INFRASTRUCTURES

At the bottom layer of the architecture are the underlying IaaS cloud infrastructures. In Figure 2, three possible infrastructures are shown: the USTRANSCOM private cloud, the Amazon GovCloud restricted public cloud, and the Defense Information Systems Agency (DISA) DoD cloud.

The current instantiation of the LSE is a private cloud and serves as a model for a potential USTRANSCOM private cloud; it also serves as the development environment Lincoln Laboratory is using to build a prototype of this next-generation architecture. This physical infrastructure must also provide a "root of trust" for the cyber security system. This root of trust is a set of functions, defined in specialized hardware, that provides security guarantees about the system. It must be possible to know that the lowest layers of the system have not been corrupted, i.e., that the boot process is free of malware. Guarantees of security at the lowest levels of the system can provide assurances at higher levels of abstraction—assurances that demonstrate that the cyber security system that crosses all layers is working as designed.

Lincoln Secure Environment

The LSE is a prototype environment designed to serve as a (1) sandbox that can be employed to test out options for secure software development and (2) an operational high-assurance, multitenant development environment that can be tested and evaluated for usability on the basis of actual development efforts. As a security sandbox, the prototype environment can be used by Lincoln Laboratory staff to investigate techniques that mitigate some of the risks of system intrusion and theft of sensitive materials. As a development environment, this prototype of a usable, secure system, with concrete requirements, designs, and implemented technology, can be transferred to USTRANSCOM and incorporated in its efforts to build a large-scale Common Computing Environment (CCE).

LSE Architecture

The LSE architecture design was driven by the CCE requirements, Lincoln Laboratory's developer requirements, and the LSE threat model. The LSE was designed to achieve the following goals:

- Provide a secure and usable environment, capable of hosting more than one group of tenants
- Provide each group of tenants with one or more secure development enclaves for their work
- Provide shared, persistent storage within an enclave, with a consistent, roaming profile for each developer
- Secure each developer enclave so that work and data are not visible from other enclaves
- Provide a rich set of shared tools, so that users have the flexibility, within an enclave, to choose their preferred tool chain for source control, build management, release management, and software assurance
- Enforce a secure "single front door" to the LSE and its enclaves
- Enforce two-factor authentication with a hardware-based token for access
- Ensure an automated, configuration-controlled environment, minimizing system administration efforts and guaranteeing a known, reviewed, tested, secured, and malware-free deployment for every element in the LSE
- Allow an evolutionary path for the LSE, so that it can grow from virtualization to IaaS and, eventually, to PaaS

This architecture maps to the lowest three layers in Figure 2. Conceptually, the LSE is a set of isolated enclaves that run on virtual and physical resources. A user sees each enclave as a collection of virtual machines that are accessed from a remote host, through a virtual firewall, as shown in Figure 3.

The LSE is partitioned into two types of enclaves:

1. The developer enclaves are isolated enclaves used to build and test software. Users can spin up virtual machines within their enclave as needed in the development process. The Lincoln Laboratory USTRANSCOM development team "lives" in one of these enclaves. A separate test enclave is used for exercising the latest builds and running the full test suite.

[4] The Lincoln Laboratory team conducted a study to assess the current state of the art for cloud technologies and the gaps at USTRANSCOM. The result of this study is a key recommendation for USTRANSCOM to move toward a hybrid cloud option, deploying to a secure government cloud wherever feasible for unclassified data.

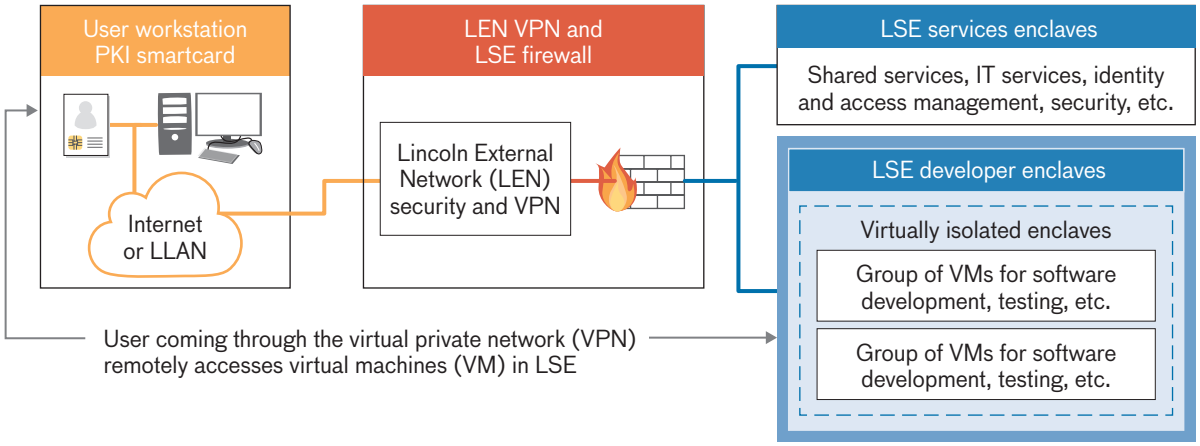


FIGURE 3. Each user has one secured path into the Lincoln Secure Environment (LSE), leading to the compartmentalized developer enclaves for each specific project. Developers can collaborate with others on their project, completely isolated from other development projects in the LSE. All developers have access to a common set of services.

2. The services enclaves provide tools and services that are shared by all developers. For example, the software services enclave houses a shared Git multitenant repository, a Nexus server, a Jenkins build server, and other development-oriented services. Common system-wide services, such as Network Time Protocol (NTP), Domain Name Service (DNS), and the like, are provided by the shared infrastructure services enclave.

Virtual firewalls implement additional traffic segmentation between enclaves within the LSE. These enclaves are also isolated from each other via hardware virtualization and network isolation techniques.

In order to promote consistency, repeatability, and configuration management, the LSE leverages automated provisioning tools to virtualize and deliver the infrastructure as a service. This automation, a key contributor to the security of the LSE, provides a scalable, repeatable, auditable method for ensuring secure configurations that are preapproved and enforced within the LSE. The LSE uses configuration-controlled Salt scripts [19] to define the environment programmatically, enabling the orchestration and interconnection of the LSE components. Other Salt scripts ensure that the security controls meet required guidelines, implementing the DoD Security Technical Implementation Guides mandatory for systems at USTRANSCOM. The collection of Salt scripts allows the system administrator to ensure correct, secured components are configured and installed. These scripts also enable system administrators to quickly wipe the entire system and recreate it in a known-good state after malware has been detected or a cyber attack has occurred. Finally, these scripts define the LSE; upgrade a script and the entire LSE is upgraded.

Threat Model

Because USTRANSCOM's critical planning information, generated plans, applications, and algorithms are valuable to U.S. adversaries, USTRANSCOM faces two important challenges to ensuring its continued ability to perform the U.S. military's logistics role.

First, the three classes of networks to which USTRANSCOM connects have varying levels of protection. The least well-protected networks are those that are directly connected to the Internet and include the networks of commercial partners that ship nonsensitive materiel. Some commercial companies provide good cyber protection, but not all companies are diligent. The second class of networks includes those used by the military and government to provide USTRANSCOM with requirements about missions.[5] The U.S. government protects these networks from cyber attack by using a blend of commercial and government-developed tools and techniques. The third class of networks consists of those within USTRANSCOM and its Transportation Component

[5] These networks may include nonmilitary government agencies, such as the Centers for Disease Control and Prevention and the Federal Emergency Management Agency, which is involved in U.S. humanitarian aid or disaster relief efforts.

Commands that host the software and data used to plan and execute logistics operations. These networks are the most sensitive and require extra protection.

Second, USTRANSCOM needs to address adversaries who have a wide range of capabilities and who represent the three classes of sophistication (each grouped into a pair of tiers) described in the Report of the Defense Science Board Task Force on Resilient Military Systems and the Advanced Cyber Threat [17]. At the lowest class are the Tier I-II practitioners who rely on others to develop malicious code. These cyber attackers are mainly nuisances, looking to attack public-facing USTRANSCOM websites to demonstrate their capabilities as "hackers." The DoD uses a variety of techniques, not unique to USTRANSCOM, to limit the access and impact of these nuisances. In the middle class are the Tier III-IV adversaries who can develop their own tools to exploit known vulnerabilities and to discover unknown existing vulnerabilities. These actors appear across all networks, internal and external, usually in search of some financial gain. The final class, Tier V-VI, comprises other great powers, who have sufficient resources to create vulnerabilities in systems and who focus their efforts on USTRANSCOM's most sensitive networks and data. These Tier V-VI adversaries are constantly seeking entry to USTRANSCOM systems as a way to gain continued access to U.S. classified information. These actors also attempt to compromise external vendor systems as a "back door" into USTRANSCOM's enterprise. To effectively execute all missions all the time, one would need to protect against the top-tier threats. However, not all data are of equal value. Clearly, data that indicate military plans are of the highest value, and information that discloses the delivery of essential materiel within short time windows, suggesting upcoming military operations, are of great importance.

Broadly speaking, USTRANSCOM needs to ensure that its commercial partners practice good authentication and authorization, perform regular "cyber hygiene" to ensure their systems are patched and up to date, leverage virus detection, and use software developed by a team of people who know how to develop code resilient to cyber attacks. USTRANSCOM's connections to and from these partners need to be done over secure channels. These precautions address the low-level Tier I-II attackers.

USTRANSCOM needs to do more to address the Tier III-IV attackers. To thwart these adversaries, techniques are needed to securely store data and to verify that data are not modified as they traverse the system. Data must be tracked by using secure provenance techniques [20], and the software should be verified by employing trusted or secure boot techniques [21]. A secure development environment like the LSE provides additional protections [22].

For the serious Tier V-VI adversaries, it must be assumed that, with their skills and persistence, they will succeed in penetrating the USTRANSCOM enterprise. Thus, the aim is to improve the detection and deterrence of attacks, effectively raising the costs for adversaries to reach their goals. This objective is difficult to achieve through technical means because adversaries only need to successfully exploit one vulnerability to gain access to the enterprise while the enterprise's cyber defenders must protect against all vulnerabilities. By coupling strong authentication and authorization with data provenance, one can better attribute certain attacks to certain actors. Knowing who is responsible can provide insights into what tactics the adversary may use next, leading to additional defenses and a stronger response. This attribution can be shared with other DoD and intelligence community cyber defenders to improve their situational awareness and defensive posture. Improving cyber defensive capabilities for thwarting Tier V-VI adversaries is an ongoing effort by the DoD, as well as in the continuing collaboration between USTRANSCOM and Lincoln Laboratory.

Security Architecture

The security requirements of the USTRANSCOM CCE form the basis of the LSE requirements. USTRANSCOM CCE requirements include those driven by the DoD. The Joint Information Environment defined by DISA is a part of the DoD's strategic plan that supplies requirements for the CCE. The Joint Information Environment defines its Single Security Architecture (SSA) with a vision for "a single joint enterprise IT [information technology] platform that can be leveraged for all DoD missions" [23]. Table 1 shows how SSA design principles are satisfied by LSE design artifacts.

The LSE implements a *least privilege* model to control access and determine how to elevate access for users. In this model, users get the smallest set of user rights and

privileges necessary for performing their work. This model is applied on a per-user–per-project basis. Users may have elevated privileges for one project, i.e., in one enclave, but not for another project in another enclave.

LSE Usability Results

A key objective of the LSE is usability; if a system is difficult to use because of security measures, users will find a way to work around the difficulties, thereby weakening security. The team developed a survey to determine three things:

- Is the LSE usable for typical development work?
- Does the LSE implementation achieve a balance between security and usability?
- Are there common elements affecting ease of use and productivity in the LSE?

The survey was created by taking demographic questions specific to users in the LSE, together with questions intended to compare the LSE environment with the users' typical work environment, and integrating the industry-standard System Usability Scale (SUS) [24] (i.e., the questions listed in Table 2). The survey took a snapshot of the work being conducted within the LSE, with the goals of (1) rating the process of bringing someone on board and into the LSE, (2) comparing software development in the LSE with Lincoln Laboratory software development outside the LSE, and (3) assessing the ease of conducting daily development tasks within the LSE. The SUS was incorporated to evaluate perceived user satisfaction with the LSE and provide a score that can be compared against a large number of industry examples. The pattern of questions in the SUS, with a positive question followed by a

Table 1. Joint Information Environment Single Security Architecture (SSA) Principles and Lincoln Secure Environment (LSE) Design Artifacts

SSA PRINCIPLE	LSE DESIGN ARTIFACTS
Resiliency	• All LSE nodes are redundant to allow services to migrate from one virtual machine to another and from one hardware host to a different one. • All the networks are separated from each other by using physical, logical, and cryptographic separation. • All the communications protocols are standard. Specific authentication mechanisms are VMware-specific because currently there is no interoperability between the vendors that support Common Access Card (CAC)-based authentication. DoD-issued CACs provide a required second factor for authentication.
Maneuverability	• LSE administrators can control and configure authentication mechanisms to address emerging needs and security conditions. • Access is granted to data and services, not servers. • Everything inside the LSE is virtualized as much as practical.
Accessibility	• Every user and every device will be authenticated. • Configuration is policy based and is enforced. • All transactions are authorized through access control.
Visibility	• All the network and service-providing hosts are monitored continuously. • All the alerts and other artifacts of the monitoring are fed to a separate network operations center for analysis and for incorporation into a shared situational awareness picture. • All the enclaves and all the nodes within each enclave will conform to the relevant network-related and host-related DoD Security Technical Implementation Guides.

negative question, was deliberately designed to reduce response bias. Respondents indicated their scores for each question on a five-point Likert scale that ranges from a low score of "strongly disagree" to a high score of "strongly agree" [25]. The survey was conducted twice: once to evaluate the initial version of the LSE and later to assess the LSE after improvements had been made.

Eleven LSE users (nine developers and two analysts) responded to the initial survey. Fourteen LSE users (thirteen developers and one analyst) responded to the second survey; a majority of these respondents had used both instantiations of the LSE. As of the second survey, nine of the thirteen developers were using the environment for a majority of their development tasks. For six of the thirteen developers, the LSE was their only development environment, with 100% of their work conducted within that environment. Figure 4 provides a census of some of the software applications the developers and analysts used in their work within the LSE.

Overall, users found that the LSE supported their tasks well (average score of 3.21 on a 5-point Likert scale) and considered the system more secure when compared to their desktop system (average score of 3.85 on the 5-point scale). Additionally, ten of the returning users reported that the system had been improved between surveys. The overall SUS score for the latest release of the LSE was 52 (compared to 44 for the first instantiation of the LSE). Figure 5 shows a comparison of the SUS scores by user between the first and the current versions of the LSE. The SUS score is generated by summing and scaling calculations that equalize the impact of each question. This nonlinear score can be normalized and compared with thousands of other SUS results. The overall mean SUS score for a wide range of open software that users consider usable is 68 [26]. It should be noted that usability scores for closed systems such as the LSE will likely never match the scores for an open system because the requisite security measures for closed systems add complexity

Table 2. System Usability Scale (SUS) Questions

NUMBER	QUESTION
Q1	I think that I would like to use this system frequently.
Q2	I found the system unnecessarily complex.
Q3	I thought the system was easy to use.
Q4	I think that I would need the support of a technical person to be able to use this system.
Q5	I found the various functions in this system were well integrated.
Q6	I thought there was too much inconsistency in this system.
Q7	I would imagine that most people would learn to use this system very quickly.
Q8	I found the system very cumbersome to use.
Q9	I felt very confident using the system.
Q10	I needed to learn a lot of things before I could get going with this system.

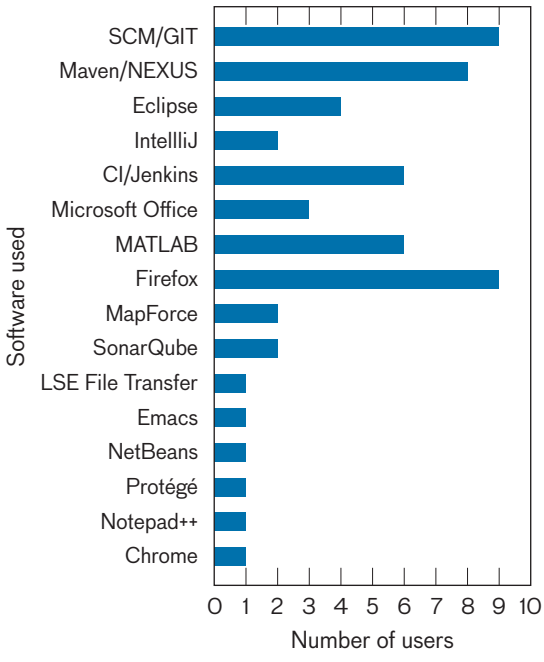


FIGURE 4. The software applications used in the Lincoln Secure Environment (LSE) by the developers and analysts who took the survey on the environment's usability show the range of programs the LSE could support.

and procedures that make a system less convenient and intuitive for operators to use. However, aiming for a high usability score assures that usability is a key consideration in the ongoing development of the LSE.

Recent system modifications have resulted in most of the users (>75%) reporting that these improvements have increased usability significantly, and gains were made to lessen the perceived need for the support of a technical person (Figure 6, Q4) and to increase reported overall confidence in system use (Figure 6, Q9). Additionally, SUS scores among respondents to the second survey are more consistent, signaling that improvements may have addressed the most impactful usability issues and that new users do not feel as though they are at a significant handicap. However, the SUS scores indicate that there is still a need for further improvement. Periodic assessment will ensure that LSE development has a continuing focus on the needs of its users, and utilizing the standard SUS scoring mechanism will ensure a fair comparison with future surveys of LSE usability.

Data Provenance in the Lincoln Secure Environment

One technique we leverage to address lower-tier and middle-tier adversaries uses data provenance, the recorded evolution of data in a system [9], to produce a set of mandatory access control (MAC) rules. The challenge in using data provenance is instrumenting systems and applications to generate and capture provenance information. USTRANSCOM has, over the years, built, grown, evolved, and acquired a rich, powerful set of legacy applications, systems, and information technology infrastructure to support its business processes. These legacy applications access, create, use, manipulate, and store transportation-related data necessary for analyzing, planning, scheduling, assembling, and executing a mission. Attempting to instrument USTRANSCOM applications to generate application context-based provenance would provide the best information, but would be expensive in time and effort. Kernel-level provenance, in which information is tracked by actions in the operating system, such as reads and writes to files and sockets, quickly and easily produces coverage of provenance events, but at a low level, without much application context.

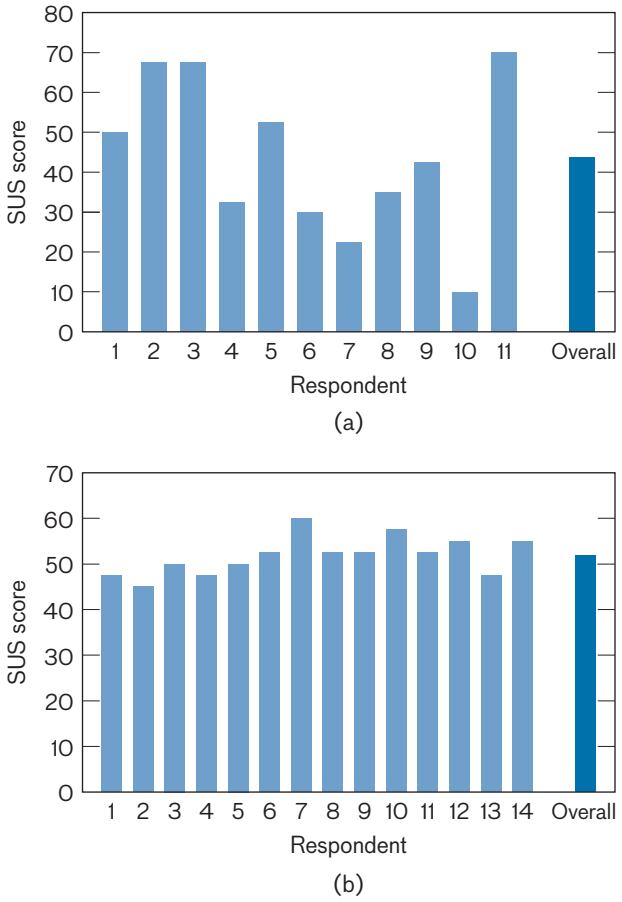


FIGURE 5. System Usability Scale (SUS) scores by users in the Lincoln Secure Environment usability surveys; the initial survey results are in (a) and post-improvements results are in (b). A score of 68 is the average usability score reported in the many usability studies conducted by industry and research institutes.

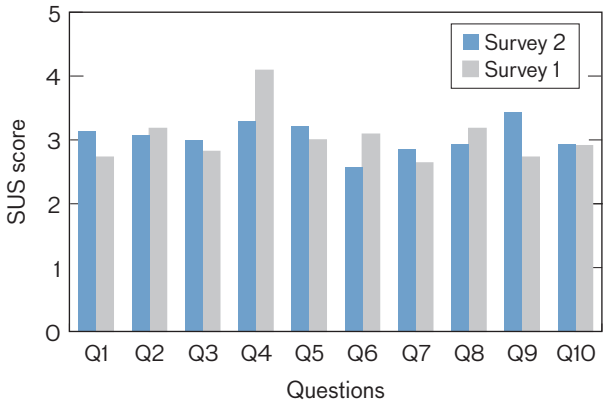


FIGURE 6. System Usability Scale (SUS) scores by question in the Lincoln Secure Environment usability survey. Significant improvements are shown in Q4 (perceived need for technical assistance) and Q9 (perceived confidence of use).

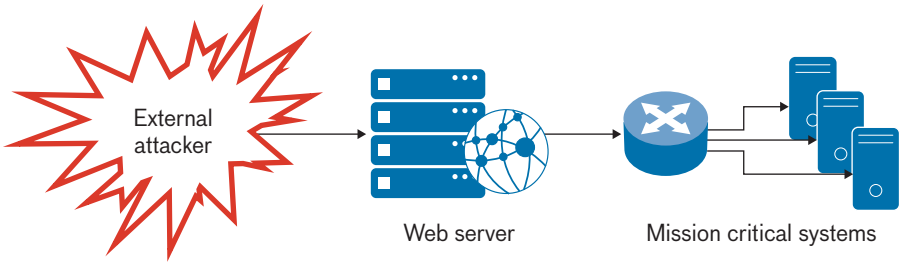


FIGURE 7. The adversary on the left accesses a targeted web server via a vulnerability in its code and then is able to gain admittance to other mission-critical systems from which the adversary can exfiltrate data.

Typically, when an adversary takes over an application through a security vulnerability, the adversary has all the rights associated with that application. For example, if the application is used to access a database whose contents should be kept confidential, then the adversary has access to the contents of that database and can exfiltrate or modify the data. Figure 7 shows an example of how an adversary can gain access into an organization's network. The cyber attacker exploits a vulnerability in the web server by introducing malicious software and consequently obtains all the rights and privileges of a system administrator assigned to the web server. The attacker can now use these privileges to access other mission-critical systems.

Efficiently collecting provenance data at the kernel level [9, 27] enables us to get a complete picture of the behaviors of subjects (e.g., users, applications, processes that request access) and objects (e.g., files, databases, computers that are accessed) in a system. In a system that collects provenance, the adversary in Figure 7 leaves a trail of evidence for detection and forensic analysis—but that trail is identified too late, and the adversary has gained access. The provenance information generated by an uncompromised system can be used to build SELinux (Security-Enhanced Linux, a version of the Linux operating system that supports access control policies) MAC policies that block the adversary from compromising the web server in the first place.

Table 3 shows a fragment of an SELinux MAC policy for the Firefox web browser. This fragment is designed to prevent an initial browser compromise, which is the first step in compromising the web server. The left and right columns show line numbers and SELinux rules, respectively. The goals of the fragment are to define a limited e4684_firefox_t domain and to ensure that users move from the unconstrained unconfined_t domain into the restricted e4684_firefox_t domain.

Manually writing such a MAC policy is difficult: many of the rules for such a policy can span multiple pages; missing a single rule can cause the application to function incorrectly; writing the policy requires the policy writer to have in-depth knowledge of system and application behavior; and often the interaction of rules within the policy are unknown. Manually developing MAC policies often results in policies that are either too restrictive or overly permissive.

The difficulties in manually developing MAC policies have driven research into several approaches that partially or fully automate the process [28, 29]. Some of these approaches record the interactions of an application with the operating system, gathering and analyzing data from that interaction to build policies. The drawback of these approaches is that gaps might exist in the data. Gaps in the data make it impossible to generate a complete set of rules, resulting in MAC policies that are excessively restrictive or too permissive. To use data provenance to build MAC policies, the provenance data for an application in a system must be complete.

Using Provenance To Expedite MAC Policies (UPTEMPO) is a Lincoln Laboratory framework that utilizes kernel-level data provenance to expedite the generation of MAC policies, thereby automating the securing of computing systems. The MAC policies that UPTEMPO builds ensure that the integrity of data is preserved and limits the damage adversaries can do when they are able to compromise an application. UPTEMPO automatically generates policies conforming to the Biba integrity model [30] by using provenance data on subjects and objects in a computing system. Figure 8 shows the five stages in UPTEMPO: (1) provenance collection, (2) policy generation, (3) policy refinement, (4) policy translation, and (5) policy enforcement.

In the first stage, UPTEMPO collects provenance data on the subjects and objects in a system and stores the data in a provenance data store. In the second stage,

UPTEMPO analyzes the provenance data and uses the results of the analysis to generate a graph representation of the final MAC policy. In the third stage, UPTEMPO refines the graph to remove redundant edges and nodes. In the fourth stage, UPTEMPO translates the refined graph into a MAC policy. Finally, in the fifth stage, UPTEMPO enforces the MAC policy. UPTEMPO addresses the problem of overly restrictive or overly permissive policies by routinely iterating through the five stages.

A production computing environment that uses UPTEMPO to generate policies protects the web server and the mission-critical systems accessed by the server. UPTEMPO policies raise the cost of an attack by slowing the adversary at every step. If the adversary manages to find an effective compromise for the web server that works around UPTEMPO's policies, additional policies protect the mission-critical systems.[6] Applications are only allowed to access the data they need to function correctly.

UPTEMPO collects provenance information as the system is used. On a regular basis, a system administrator would use UPTEMPO to incrementally generate an updated set of MAC policies. Initially, the human in the loop would be responsible for generating the policies. In the longer term, this function could be an automated, regular occurrence, removing the human except as someone to sanity check the results. This recurrent policy updating improves security by denying adversaries the time to construct workarounds to MAC policies.

[6] This technique, known as "defense in depth," uses defenses (or walls) that are not just around the perimeter of the system. Because we expect top-tier adversaries to get inside those walls, we build more defenses inside.

UPTEMPO Evaluation

A common attack scenario relies on a compromised web server and a vulnerable program that visits that web server. When a vulnerable program visits the compromised web server, the program is then compromised. Through this compromised program, the adversary subsequently gains access to mission-critical systems in order to exfiltrate data stored there.

To demonstrate the feasibility of the UPTEMPO approach, we considered the Firefox web browser as a proxy for a mission-critical application and utilized user files as the exfiltrated data. The goal of this initial evaluation was to demonstrate the use of UPTEMPO to restrict a compromised Firefox browser's functionality to web browsing only, thus prohibiting an adversary from accessing user files.

We followed a four-step process for this evaluation:

Step 1: Collect provenance data from an uncompromised Firefox browser.

Table 3. An Example Fragment of an SELinux Policy for the Firefox Web Browser

LINE	SELINUX RULE
1	allow e4684_firefox_t NetworkManager_t:dbus send_msg;
2	allow e4684_firefox_t bin_t:dir {read search open getattr};
3	allow e4684_firefox_t bin_t:file {read execute open getattr};
4	allow e4684_firefox_t bin_t:lnk_file {read getattr};
5	allow e4684_firefox_t bin_t:unix_stream_socket connectto;
6	allow e4684_firefox_t config_home_t:dir {write remove_name search add_name};
7	allow e4684_firefox_t config_home_t:file {rename write getattr read create unlink open};
8	allow e4684_firefox_t d2799_je23930_t:dir {read search open getattr};
9	allow e4684_firefox_t dns_port_t:udp_socket name_connect;
10	type_transition unconfined_t e4684_firefox_exec_t:process e4684_firefox_t;

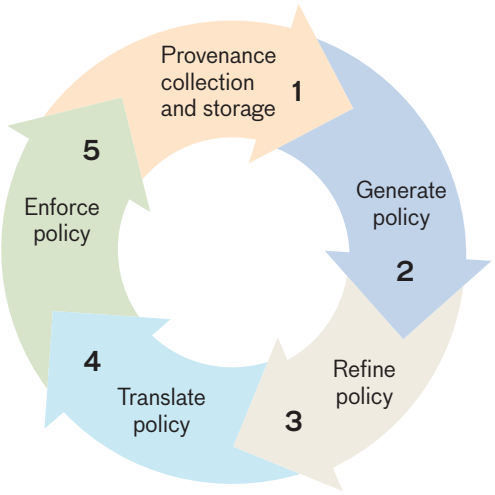


FIGURE 8. The UPTEMPO process is a continual cycle of policy generation and enforcement.

Step 2: Use UPTEMPO and the provenance data collected in Step 1 to create an SELinux policy for limiting the use of Firefox.

Step 3: Disable SELinux on the system and start Firefox. We verified that Firefox was able to browse the web and read user files.

Step 4: Enable SELinux on the system. SELinux, by default, denies all accesses to objects by subjects on a system. Without the UPTEMPO policies in place, Firefox would not be able to browse the web or read user files. We verified that the policies constructed by UPTEMPO allowed Firefox to browse the web but prevented any attempt to read user files.

UPTEMPO generated 246 types, classes, and rules for the policy constraining Firefox. The case study evaluation outlined above exercised 173 of those 246 types, classes, and rules. This initial evaluation shows UPTEMPO has great promise as a tool to further secure an environment. The next steps for this work are (1) more rigorous evaluation of the algorithms and techniques utilized to generate the policy; (2) the development of automated testing procedures that provide extensive coverage of policy cases and, when augmented by clever human-designed tests, result in a thorough assessment of the validity of these complex policies; and (3) automated provisioning of SELinux VMs in the LSE. The LSE does not currently support SELinux, which limited the integration of UPTEMPO into the LSE.

Future Directions

Enhancing the effectiveness and cyber security of USTRANSCOM's enterprise is a significant ongoing effort that requires interaction among a diverse set of organizations at various levels. Collaboration between Lincoln Laboratory and USTRANSCOM, primarily at the research and development level, is a key driver for advancing the cyber security of the more than 70 USTRANSCOM Programs of Record. LSE, for example, has been transitioned to USTRANSCOM and is being used as a template for the development of its Common Computing Environment (CCE).

Research and development efforts must be continued to ensure improved cyber security of the USTRANSCOM enterprise as it moves from a largely private, stove-piped infrastructure to a unified, cloud infrastructure. This transition should employ advanced technology, such as moving target techniques, sophisticated key management, and heightened provenance collection and mining, at all levels of the application stack.

The continuing development and use of the LSE is providing insights into a secure, usable development environment; these activities also offer a valuable foundation for future work. In 2012, the DoD issued a directive to transfer computing into the cloud where feasible [31]. Moving the LSE into the cloud would help USTRANSCOM understand the implications of moving their CCE to the cloud. The automated provisioning of the LSE should make a transfer to the cloud relatively simple. The cloud can deliver additional capabilities because of its scalability, and the LSE would no longer be limited by its current hardware. Developers using the LSE in the cloud would furnish key SUS usability metrics that could be compared with the SUS scores already gathered. Securing the LSE in the cloud could leverage other cloud cyber security work being done at Lincoln Laboratory.

Building the LSE with an integrated UPTEMPO will generate an environment with improved security, but much more provenance work remains to be done to assure this improvement. An enhanced UPTEMPO could also be used as the first pass at understanding the security properties of the planning and analysis tools being developed at Lincoln Laboratory as part of the USTRANSCOM project.

These tools, which are designed to test new planning algorithms for robustness and for plan resiliency, read from databases and generate new plans. All of these sensitive

data and plans are sought by adversaries. A clever adversary in the environment might act subtly; for example, tampering with data to add a few delays in the transport of essential men and materiel could be far more effective in hampering a U.S. mission than crashing an entire system might be. Protecting data means keeping them confidential and guaranteeing their integrity. Therefore, understanding effective, efficient ways to cryptographically protect those data at rest, in motion, and in use is an issue Lincoln Laboratory is looking at on other projects. A practical USTRANSCOM set of applications and data provides researchers with valuable real-world examples and requirements that can be shared with other Laboratory projects.

In addition, determining the application-level provenance of the data in these tools can yield new cyber security insights. UPTEMPO provides a wealth of low-level provenance information. Adding application-level provenance to the tools can supply much better context to the use of data by the tools. Collecting the application provenance and merging it with the kernel-level provenance from UPTEMPO to explore provenance's potential implications to cyber security is another exciting area of research.

The partnership between USTRANSCOM and Lincoln Laboratory has produced practical prototypes and transferrable technology that will have value to programs beyond USTRANSCOM's. The collaboration is generating additional important questions, and, with both the LSE and UPTEMPO, a foundation is already available for investigating those interesting new questions. ■

References

1. Wikipedia, "Hannibal's Crossing of the Alps," available at https://en.wikipedia.org/wiki/Hannibal%27s_crossing_of_the_Alps.
2. Wikipedia, "Battle of the Atlantic," available at https://en.wikipedia.org/wiki/Battle_of_the_Atlantic.
3. "Why Germany Really Lost World War II," *The European Union Times*, posted 22 Jan. 2010, available at http://www.eutimes.net/2010/01/why-germany-really-lost-world-war-ii/.
4. United States Transportation Command website, "About USTRANSCOM," available at http://www.ustranscom.mil/cmd/aboutustc.cfm.
5. United States Transportation Command, Publication P35-1 "United States Transportation Command," available at http://www.ustranscom.mil/cmd/fpindex.cfm.
6. T. Rorabaugh, "Evaluating the Use of Cloud Hosting for Consolidating USTC IT Infrastructure," The MITRE Corporation, 2014.
7. USTRANSCOM, "AT21 [Agile Transportation for the 21st Century] Time-Phased Optimization Capabilities Storyboard White Paper Air Centric (Draft)," TCAC-F/TCAC-SL, 13 Sept. 2011.
8. USTRANSCOM, "AT21 Time-Phased Optimization Capabilities Storyboard White Paper Deployment Sealift Centric (Draft)," TCAC-F/SDTE-ST, 5 Dec. 2011.
9. D.J. Pohly, S. McLaughlin, P. McDaniel, and K. Butler, "Hi-Fi: Collecting High-Fidelity Whole-System Provenance," *Proceedings of the 28th Annual Computer Security Applications Conference*, 2012, pp. 259–268.
10. J.P. Anderson, "Computer Security Technology Planning Study," Volume II of the Report of the Computer Security Technology Planning Study Panel, 1972.
11. Defense Information Systems Agency, Identity and Access Management (IDAM) webpage, accessed Dec. 2015, available at http://www.disa.mil/Initiatives/Identity-Access-Mgmt.
12. D. Staheli, V.F. Mancuso, M.J. Leahy, and M.M. Kalke, "Cloudbreak: Answering the Challenges of Cyber Command and Control," *Lincoln Laboratory Journal*, vol. 22, 2016, pp. 60–78.
13. W. Moser, "Design and Development of the TFDM Information Management Architecture," *Proceedings of the Integrated Communications, Navigation and Surveillance Conference*, 2009, pp. 1–12.
14. V. Mehta, S. Campbell, J. Kuchar, W. Moser, H. Reynolds, T. Reynolds, and R. Seater, "The Tower Flight Data Manager Prototype System," *Proceedings of the 30th IEEE/AIAA Digital Avionics Systems Conference*, 2011, pp. 2C5-1–2C5-15.
15. K. Claypool, "Common Support Services Information Management," MIT Lincoln Laboratory Air Traffic Control Workshop, Washington, D.C., Dec. 2012, available at http://www.ll.mit.edu/mission/aviation/publications/publication-files/ms-papers/Claypool_2012_LLATC_MS-72923_WW-26438.pdf.

16. C. Kelly, "CIWS Data Distribution Service," MIT Lincoln Laboratory Air Traffic Control Workshop, Washington, D.C., Dec. 2012, available at http://www.ll.mit.edu/mission/aviation/publications/publication-files/ms-papers/Kelly_2012_LLATC_MS-72864_WW-26438.pdf.
17. Department of Defense, Report of the Defense Science Board Task Force on Resilient Military Systems and the Advanced Cyber Threat, J.R. Gosler and L. Von Thaer, task force cochairs, Jan. 2013.
18. N.A. Schear, P.T. Cable, R.K. Cunningham, V.N. Gadepally, T.M. Moyer, and A.B. Yerukhimovich, "Secure and Resilient Cloud Computing for the Department of Defense," *Lincoln Laboratory Journal*, vol. 22, 2016, pp. 123–135.
19. SaltStack, SaltStack Automation for CloudOps, ITOps and DevOps at Scale, available at http://saltstack.com.
20. A. Bates, D. Tian, K.R.B. Butler, and T. Moyer, "Trustworthy Whole-System Provenance for the Linux Kernel," *Proceedings of the 24th USENIX Security Symposium*, 2015, pp. 319–334.
21. R. Sailer, X. Zhang, T. Jaeger, and L. van Doorn, "Design and Implementation of a TCG-Based Integrity Measurement Architecture," *Proceedings of the 13th Conference on USENIX Security Symposium*, vol. 13, 2004, p. 16.
22. Department of Defense, Report of the Defense Science Board Task Force on Cyber-Security and Reliability in a Digital Cloud, E.D. Evans and R. L. Grossman, task force cochairs, Sept. 2012.
23. Defense Information Systems Agency, "Shaping the Enterprise for the Conflicts of Tomorrow," 2014, available at http://www.disa.mil/~/media/Files/DISA/About/JIE101_000.pdf.
24. J. Brooke, "SUS: A Quick and Dirty Usability Scale," Chapter 21 in *Usability Evaluation in Industry*, P.W. Jordan, B. Thomas, B.A. Weerdmeester, and I.L. McLelland, eds. Bristol, Penn.: Taylor & Francis, 1996.
25. R. Likert, "A Technique for the Measurement of Attitudes," *Archives of Psychology*, vol. 22, no. 140, 1932, pp. 1–55.
26. J. Sauro, "Measuring Usability with the System Usability Scale," *Measuring Usability* website, 2 Feb. 2011, available at http://www.measuringu.com/sus.php.
27. A. Bates, K.R.B. Butler, and T. Moyer, "Linux Provenance Modules: Secure Provenance Collection for the Linux Kernel," Technical Report CIS-TR-201407-01, University of Oregon, 2014.
28. N. Provos, "Improving Host Security with System Call Policies," *Proceedings of the 12th Conference on USENIX Security Symposium*, vol. 12, 2003, p. 18.
29. B.T. Sniffen, D.R. Harris, and J.D. Ramsdell, "Guided Policy Generation for Application Authors," MITRE Technical Papers, The MITRE Corporation, 2006.
30. K.J. Biba, "Integrity Considerations for Secure Computer Systems," The MITRE Corporation, Technical Report 76-372, 1977.
31. "Cloud Computing Strategy," Report issued by the Chief Information Officer, U.S. Department of Defense, 5 July 2012.

About the Authors

Jeffrey M. Diewald is a member of the technical staff in the Secure Resilient Systems and Technology Group at Lincoln Laboratory. Since joining the Laboratory in 2011, he has been working on the development of secure, usable software for building and managing certificate authorities, provenance, data integrity, and dynamic key management. His recent efforts are in helping to add provenance technologies to the USTRANSCOM project code base, to bring usable security to the Lincoln Secure Environment, and to understand how cloud technologies can help USTRANSCOM. Previously, he worked for several notable companies, including Digital Equipment Corporation, Compuware, and Avid Technology. During that time, he developed and delivered compilers, debuggers (resulting in two U.S. patents), software performance tools, and user interfaces for several significant products. He holds bachelor's degrees in computer engineering and electrical engineering, and a master's degree in electrical engineering, all from the University of Michigan.

Kajal T. Claypool is currently the assistant leader for the Informatics and Decision Support Group and the program lead for Lincoln Laboratory's USTRANSCOM project. She has also worked on several of Federal Aviation Administration's Next-Generation Air Transportation System programs during her tenure at Lincoln Laboratory and is actively engaged in research and technology development in big data management. She has been active in the database research community for more than 15 years, and her interests include information integration, information retrieval, data analytics, and secure architectures. She has authored numerous publications in these and related areas. Prior to joining the Laboratory in 2007, she worked as a member of technical staff at Oracle Corporation and served as an assistant professor in the Department of Computer Science at the University of Massachusetts, Lowell. She holds a bachelor of technology degree in computer engineering from the Manipal Institute of Technology, Karnataka, India, and a doctoral degree in computer science from Worcester Polytechnic Institute.

Jesslyn D. Alekseyev is a member of the technical staff in the Informatics and Decision Support Group. She is currently a human-factors researcher and user-experience designer on the USTRANSCOM and Biosurveillance Ecosystem projects. Her work at Lincoln Laboratory has focused on interface design, information architecture, and user research and analysis. Prior to joining the Laboratory, she worked as a designer, with an interest in process and resource management for exhibit, environmental, and wayfinding projects

and with additional experience in transportation logistics through her work with the United Parcel Service. She has a bachelor's degree in illustration from Syracuse University, with additional coursework in communication design, interaction design, and psychology. She is currently working toward a master's degree in human factors in information design at Bentley University.

George K. Baah is a technical staff member in the Cyber Analytics and Decision Systems Group. Prior to joining Lincoln Laboratory in 2013, he was a postdoctoral fellow at the Georgia Institute of Technology. His research interests include cyber security, program analysis, machine learning, and causal analysis. He holds a bachelor's degree (magna cum laude) in computer science, with a minor in mathematics, from Pace University and a doctoral degree in computer science from the Georgia Institute of Technology.

Uri Blumenthal is a member of the technical staff in the Secure Resilient Systems and Technology Group. Since joining Lincoln Laboratory in 2007, he has worked on secure communication protocols, computer applications, and infrastructures. His professional activities have focused on information assurance, cyber assessments, identity management and access control in complex configurations, and distributed security architecture. Other areas of interest include cryptographic algorithms and protocols, network management, and theory of music. Prior to joining the Laboratory, he worked at several research facilities, including IBM Research, Bell Labs, and Intel Research Labs. He has published papers and books on cryptography, public key infrastructure, network management, and cyber situational awareness. He holds a master's degree in applied mathematics from Odessa State University and a master's degree in divinity from The Interfaith Seminary of New York.

Alfred Cilcius is a member of the Informatics and Decision Support Group. Since joining Lincoln Laboratory in 2013, he has been supporting the USTRANSCOM project by working on the Lincoln Secure Environment, developing DevOps (methodology derived from a collaboration between operations staff and development engineers), automated provisioning, secure network and enclave isolation, and open-source software. Before joining the Laboratory, he was the chief software architect at AgilePath and worked for other notable companies, including TYBRIN Corporation (now a segment of Jacobs Technology), The MITRE Corporation, Wang Laboratories, and various startups. He holds a bachelor's degree in physics from Bates College and a master's degree in systems and advanced technology from the State University of New York, Binghamton.

Joseph A. Cooley is a member of the technical staff in the Informatics and Decision Support Group. Since joining Lincoln Laboratory in 2000, he has worked on a wide range of computer system problems, including durable archival storage, distributed systems, Internet protocol networking in disadvantaged environments, applied cryptography, enterprise security, and energy systems. As a Lincoln Scholar from 2009 to 2011, he completed a master's degree in computer science at Dartmouth College under the direction of Dr. Sean W. Smith. His graduate research focused on improving usable security in the domain of medical informatics. He also holds a bachelor's degree in computer science from the University of Minnesota.

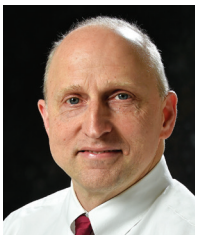
Robert K. Cunningham is the leader of the Secure Resilient Systems and Technology Group. He is responsible for initiating and managing research and development programs in information assurance and computer and platform security. His early research at Lincoln Laboratory focused on machine learning, digital image processing, and image and video understanding. As part of this effort, he contributed to early drafts of the real-time message passing interface (MPI/RT) specification. Later, as a member of the technical staff in the Information Systems Technology Group, he pursued system security research and development, initially investigating intrusion-detection systems that do not require advance knowledge of the method of attack, then moving on to consider detection and analysis of malicious software. He has patented security-related technology, presented and published widely, and served as general chair or program chair for many conferences and workshops. He has also served on several national panels, such as the U.S. Army Cyber Materiel Development Strategy Review Panel, and led national teams, such as the National Security Agency's working group for computer network defense research and technology transition. He holds a bachelor's degree in computer engineering from Brown University, a master's degree in electrical engineering from Boston University, and a doctorate in cognitive and neural systems from Boston University.

Jonathan R. Glennie is an IT systems administrator for the Air Traffic Control Systems and the Informatics and Decision Support Groups. He began working at Lincoln Laboratory in 2006 after earning a bachelor's degree in computer networking and systems administration from Michigan Technological University. Prior to joining the Laboratory, he focused primarily on Windows desktop and datacenter technologies. After being introduced to VMware virtualization in 2010, he quickly became familiar with the architecture and administration of the key components and served a central

role in expanding VMware use within the groups he supports. In 2013, he helped set up the Homeland Protection and Air Traffic Control Division's multitenant virtualization infrastructure that is used by all the groups in the division, and he now serves as the lead administrator for the infrastructure. His current work focuses on virtualization technologies for the data center and desktop, data center computing, and network engineering with an emphasis on automation techniques for today's cloud environments.

Edward F. Griffin is a system engineer in the Air Traffic Control Systems Group. He joined Lincoln Laboratory in 1998 after receiving a bachelor's degree in engineering from the University of Southern California. His focuses are on the design and implementation of data center networking and on computing and storage technologies to facilitate the operation and reliability of the various programs in the Air Traffic Control Systems and the Informatics and Decision Support Groups. He has contributed to many Laboratory-wide initiatives and committees, helping to shape the direction of computing and networking in the Laboratory. Using the skills derived through these activities, he assisted in the design and implementation of the technology used in the Lincoln Secure Environment.

William L. Pughe is a member of technical staff in the Informatics and Decision Support Group. He joined the Weather Sensing Group at Lincoln Laboratory in 1996 as a software developer working on weather-detection algorithms for the Terminal Doppler Weather Radar and Weather Systems Processor programs. Since switching to the Informatics and Decision Support Group, he has been involved in developing the Lincoln Secure Environment, a software-development test bed, and in investigating cyber security issues for industrial control systems. He holds a bachelor's degree in physics from the University of Massachusetts, Amherst.



Patrick J. Pawlak is a member of the technical staff in the Air Traffic Control Systems Group. His focus is on the development of robust networks that meet the group's demands for large, externally facing, real-time prototypes that enable the validation of technology in operational settings. Since joining Lincoln Laboratory in 1988, he has been a key member of several teams responsible for the transfer of experimental systems to the Federal Aviation Administration (FAA) for transition to an operational capability. He is currently part of the teams developing real-time prototypes for the Corridor Integrated Weather System, Consolidated Storm Prediction for Aviation, and National Weather Processor programs. He leads the team responsible for systems and network design and management for the Air Traffic Control Systems and the Informatics and Decision Support Groups. His team has applied the techniques used in supporting the FAA programs to the design and setup of the Lincoln Secure Environment. He holds a bachelor's degree in computer science from Saginaw Valley State University.